#KLOOIENMETCOMPUTERS

ARNOUT VAN KEMPEN OVER ROMMELEN IN EEN DIGITALE WERELD

Wie zoals ik is opgegroeid met MS/DOS, ervaart Linux enerzijds als een feest der herkenning. Veel van de basisconcepten zijn hetzelfde. Maar anderzijds is Linux echt overweldigend. Het biedt zoveel meer mogelijkheden, de basis directorystructuur is complexer, de hoeveelheid beschikbare commando's is enorm. En dan laat ik nog buiten beschouwing de variatie in GUI's en CLI's. U weet nog wel: grafische *interfaces* en *command line interfaces*.

Ik heb al aangegeven dat ik met een Raspberry Pi werk, met Raspian als Linux-versie. De GUI laat ik voorlopig links liggen. Dat betekent dat we de CLI gaan verkennen en dat is in deze standaardset BASH. De naam BASH is een grapje, van het soort waar UNIX, en dus ook LINUX vol mee zit. In UNIX had je de *shell*, ofwel SH, als CLI. Een van de mogelijkheden in UNIX was de shell geschreven door de heer Bourne, de Bourne-shell. In Linux is dat de basis geweest voor BASH, de Born Again Shell.

Bij het opstarten van je Pi in CLI-modus start BASH automatisch. Een van de eerste taken van BASH is het inrichten van een aantal standaardinstellingen voor alle gebruikers en voor de gebruiker die op dat moment actief is. Daar zit meteen een belangrijk punt. We zijn gewend geraakt aan personal computers die wel *multi-tasking* zijn, maar *single-user*. Op het kantoornetwerk loggen we eerst in op het singel-user MS Windows, om pas daarna contact te leggen met achterliggende servers. Linux is fundamenteel multi-tasking en multi-user. Hoe simpel zo'n Raspberry Pi ook is, in principe kan het systeem een reeks gebruikers tegelijk bedienen. Op dit moment nog niet van belang, maar het komt nog wel eens terug.

Linux werkt met directories, net zoals vroeger MS/DOS en MS Windows. Op enig moment ging Microsoft die structuren *folders* noemen, maar feitelijk is het verschil met een directory minimaal. Wat in Linux wel van belang is, is te begrijpen dat een directory een bestand, of een file, is. Ook programma's zijn files. In Windows lijken folders en files iets totaal verschillends. Programma's en overige files lijken wel op elkaar, maar in Windows kan je het verschil zien aan de extensie, de laatste drie letters van de bestandsnaam. Als de laatste drie letters na de punt 'exe' of 'com' zijn, dan zijn het programma's en zal Windows ze als programma proberen uit te voeren, wat de inhoud ook is. Folders onderscheiden zich niet door de extensie van andere files, maar door een typeaanduiding. Deze wat rommelige organisatie is het resultaat van erfenissen van MS/DOS en pogingen het systeem voor de gebruiker zo overzichtelijk mogelijk te maken.

In Linux is het allemaal een stuk systematischer, maar voor Windowsgebruikers daardoor ook echt wennen. Het verschil tussen soorten files wordt in Linux volledig bepaald door 'vlaggen'. Iedere file heeft een aantal *permission-flags* die bepalen wat je met een file kan doen. En met een andere vlag bepaal je of een file een directory is of niet. Een van de *permissions* is *execute*. Een file met die permission is dus een programma en kan worden uitgevoerd. Extensies zoals in Windows zijn daarmee overbodig en worden in de praktijk ook maar beperkt gebruikt.
Een goed begrip van permissions is van vitaal belang als we aan het programmeren gaan. Je kunt behoorlijk gek worden van een programma dat dienst weigert, tot je je realiseert dat het niet de juiste permissions heeft. Daar komen we op terug.

Voor nu de suggestie om eens wat rond te kijken. Hiervoor is het goed te weten dat bijna alle Linux-commando's een eigen hulppagina hebben die je oproept door **man commando**, of je gebruikt **commando -h**.

Een eerste commando waar je niet buiten kan is **ls**. Probeer het eens uit, zoek met **man ls** of met **ls -h** uit wat het allemaal kan. Probeer daarna ook **cd**. Je kunt geen schade aanrichten, het is puur rondkijken in je systeem. ←